Defocus-Map Estimation From Single Image

Digital Image Processing, Monsoon '23
Team Detectives

Kyrylo Shyvam Kumar	Anushka Agarwal	Vinkesh Bansal	Tejas Srivastava
2021000000	2021102023	2021102020	2021102017

Contents

1 Introduction

In computer vision and image processing, obtaining a defocus map from a single image plays a crucial role in addressing the challenges associated with depth-of-field variations. The defocus map provides valuable information about the varying degrees of blur across different regions within an image, offering insights into the scene's depth structure and focusing conditions. By discerning the levels of defocus, one can effectively distinguish between in-focus and out-of-focus areas, enabling applications such as automatic focusing, image enhancement, and depth estimation. This map is instrumental in computational photography, where the creation of aesthetically pleasing images or the extraction of three-dimensional information from two-dimensional scenes is paramount. The defocus map serves as a foundational element in numerous computer vision applications, contributing to advancements in photography, robotics, augmented reality, and various domains where accurate depth information is essential for informed decision-making and visual understanding.

Existing defocus estimation methods can be categorized into two classes: multiple images-based methods and single image-based methods. The former usually use a set of images captured by multiple camera focus settings, then defocus estimation is finished by a machine learning process. The latter mainly estimate accurate defocus at the edge locations, then the full defocus map is obtained using a propagation method. The multiple images-based methods have some limitations in practical applications due to their suffering from occlusion problem and requirement of a scene to be static. In contrast, defocus recovery from only a single image is more practical

We do a comparitive study of 2 papers that present methods to obtain a full defocus map from a single defocussed image. Both papers follow the similar idea of obtaining a sparse defocus map first using some kind of spectrum contrast, which is then propogated to the whole image using *matting.*

2 Defocus Map Estimation from a single image via Spectrum Contrast

2.1 Introduction

Due to the fact that the amount of blur can be estimated reliably only in areas of an image that has significant frequency content, we focus on the spectrum contrast at edge locations to estimate defocus blur amount and then obtain the full defocus map by a blur propagation process. In particular, we take into account the chromatic aberration caused by wavelength-dependent variation of the index of refraction of lens to reduce the error of edge detection. In order to remove high-frequency noise and texture ambiguity, a Gaussian filter with small kernel is used before calculating spectrum contrast

2.2 Methodology

1. **Defocus Model**:

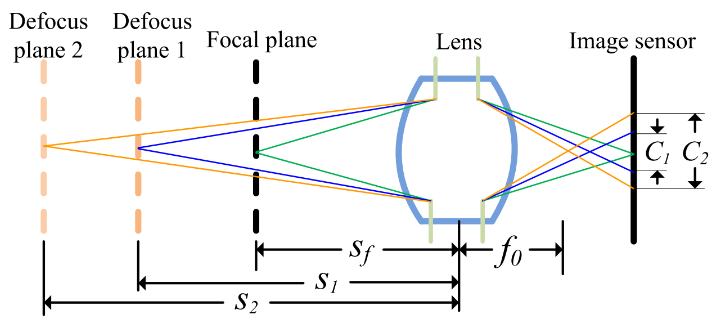

Figure 1: Thin-Lens Model

Since we assume thin lense model, all rays from the focal plane of the lens will converge to sharp point on the sensor. However, rays from nearby focal planes will also add their blurred components to that point. Thus, we can model the defocus image with a PSF as

$$d(X;\lambda) = f(X;\lambda) * p(X;\lambda) * s(\lambda) \tag{1}$$

Where $d(X;\lambda)$ is the defocused image, $f(X;\lambda)$ the ideal focused image, $p(X;\lambda)$ the PSF, and $s(\lambda)$ the wavelength sensitivity function. The PSF $p(X;\lambda)$ is usually approximated by a Gaussian function $g(x;y;\lambda;\sigma)$ and its blur Gaussian kernel σ is the defocus value which we need to estimate. $d(X;\lambda)$ is the final defocus image captured by camera

2. **Noise Removal**: We apply a small gaussian kernel to remove the high-frequency noise and supress the blur caused by bour texture.

$$g(x,y;\sigma_1) = \frac{1}{2\pi\sigma_1^2} exp(-\frac{x^2+y^2}{2\sigma_1^2})$$

3. **Chromatic Abbreations**: Due to chromatic abbreation, while extracting edges, we get three edge-maps for the three channels. In order to reduce the error of edge detection, we perform edge detection on each color channel independently and only keep edges that are detected within five pixels of each other in channel R, G, and B. Our $d(X)$ now becomes,

$$d(X) = f(X) * g(X, \sigma + \sigma_1)$$

4. **Spectrum Contrast**: We define our spectrum contrast as the absolute value of spectrum amplitude difference between one pixel and its adjacent pixels by allowing for the log spectrum representation, given as follows:

$$C(i) = ||log(||A(i)||) - \frac{1}{N}\sum_{j\in B} log(||A(j)||)||$$

where A_i represents the spectrum amplitude of pixel i, B is the neighborhood of pixel i given by the size of N, we set N 3×3 in our experiment

$$C(i) = ||log(||F(u,v)G(u,v;\sigma(i)+\sigma_1)|| - \frac{1}{N}\sum_{j \in B} log(||F(u,v)G(u,v:\sigma(j)+\sigma_1)||))|| \tag{2}$$

Through the nonlinear regression analysis of statistical mathematics, the relationship between σ_i and C_i can be approximated as

$$\sigma(i) = 1/\sqrt{exp(c(i) - \sigma_1^2}$$

The σ_i values at the edge locations form a sparse defocus map $m(i)$

5. **Joint Bilateral Filtering:** In order to suppress the influence of noisy and weak edges, we apply joint bilateral filtering (JBF) to our sparse defocus map at edge locations. The JBF can rectify some inaccurate defocus values by using their adjacent defocus values along the edge

6. **Matting :** We now use a propagation algorithm (explained in section 3) to extend the sparse defocus map to the whole image to obtain the full-defocus map.

3 Defocus Map estimation from a single image

3.1 Introduction

The algorithm used in this paper is quite similar to the previous paper, the only difference being the defocus model assumed and thus, the method of estimation of the sparse defocus map. The input image is re-blurred using a known Gaussian blur kernel and the ratio between the gradients of input and re-blurred images is calculated. It can be shown that the blur amount at edge locations can be derived from the ratio. We then formulate the blur propagation as an optimization problem. By solving the optimization problem, we finally obtain a full defocus map.

3.2 Methodology

1. **Defocus Model :** We can model an ideal edge as

$$f(x) = Au(x) + B \quad \text{where } u(x) \text{ is step function}$$

Similar to the previous paper, the defocus blur can be modelled as convolution with a point spread function (PSF), approximated by a gaussian $g(x, \sigma)$ where $\sigma = k \cdot c$ measures the defocus blur amount and is proportional to the diameter of the CoC c. The blurred edge $i(x)$ is then

$$i(x) = f(x) * g(x, \sigma)$$

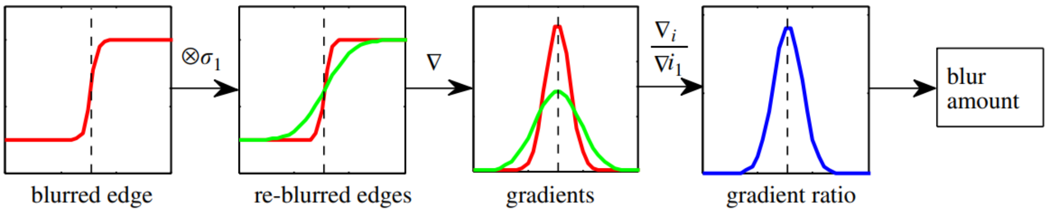

Figure 2: Defocus Blur Estimation

2. **Defocus Blur Estimation:** We re-blur the edges using another gaussian kernel, and calculate the ratio of the gradient magnitude of the step edge and its reblurred version. Since this ratio maximizes at edges, we can get defocus blur at edge locations. The gradient of it is calculated as

$$\begin{aligned}\nabla i_1(x) &= \nabla\left(i(x) \otimes g\left(x, \sigma_0\right)\right) = \nabla\left(\left(Au(x)+B\right) \otimes g(x,\sigma) \otimes g\left(x, \sigma_0\right)\right) \\ &= \frac{A}{\sqrt{2\pi\left(\sigma^2+\sigma_0^2\right)}} \exp\left(-\frac{x^2}{2\left(\sigma^2+\sigma_0^2\right)}\right),\end{aligned} \tag{3}$$

Then, the gradient magnitude ratio between the original and re-blurred edges is

$$\frac{|\nabla i(x)|}{|\nabla i_1(x)|} = \sqrt{\frac{\sigma^2+\sigma_0^2}{\sigma^2}} \exp\left(-\left(\frac{x^2}{2\sigma^2} - \frac{x^2}{2\left(\sigma^2+\sigma_0^2\right)}\right)\right) \tag{4}$$

Where the maximum value at edge locations ($x = 0$) is given by

$$R = \frac{|\nabla i(0)|}{|\nabla i_1(0)|} = \sqrt{\frac{\sigma^2+\sigma_0^2}{\sigma^2}} \quad \text{or} \quad \sigma = \frac{\sigma_0}{\sqrt{R^2-1}}.$$

3. **Joint Bilateral Filtering:** To remove effect of weak and noisy edges, we again use joint bilateral filtering on the sparse defocus map. By using the original input image I as the reference, the filtered sparse defocus map can be defined as

$$BF(\hat{d}(x)) = \frac{1}{W(x)} \sum_{y \in \mathcal{N}(x)} G_{\sigma_s}(\|x-y\|) G_{\sigma_r}(\|I(x)-I(j)\|) \hat{d}(y)$$

4. **Matting :** We now use a propagation algorithm (explained in section 3) to extend the sparse defocus map to the whole image to obtain the full-defocus map.

4 A closed form solution to natural Image Matting

We now explain the optimization step of the defocus estimation where we propagate the sparse defocus map to the whole image using *matting*. The paper introduces a novel closed-form solution for natural image matting. This solution is based on local smoothness assumptions and allows for the analytical elimination of foreground and background colors, leading to a quadratic cost function in alpha. The method simplifies the matting process and can produce high-quality mattes with minimal user input.

4.1 Methodology

The color of the ith pixel is assumed to be a linear combination of the corresponding foreground and background colors

$$I_i = \alpha_1 F_i + (1 - \alpha_1) B_i$$

We derive a cost function from local smoothness assumptions on foreground and background colors F and B and show that in the resulting expression, it is possible to analytically eliminate F and B, yielding a quadratic cost function in . The alpha matte produced by our method is the global optimum of this cost function, which may be obtained by solving a sparse linear system.

Assuming F and B are constant over small window w, we can write

$$\alpha_i \approx a I_i + b \quad \forall i \in w$$

This relation suggests finding , a, and b that minimize the cost function

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - a_j I_i - b_j)^2 + \epsilon a_j^2 \right), \tag{5}$$

The cost function above includes a regularization term on a to tackel numerical instability. We now, eliminate the unknowns a and b to get a *quadratic* equation in α

Theorem : Theorem 1. Define $J(\alpha)$ as

$$J(\alpha) = \min_{a,b} J(\alpha, a, b).$$

Then,

$$J(\alpha) = \alpha^T L \alpha,$$

where L is an $N \times N$ matrix, whose (i, j) th entry is

$$\sum_{k|(i,j) \in w_k} \left(\delta_{ij} - \frac{1}{|w_k|} \left(1 + \frac{1}{\frac{\epsilon}{|w_k|} + \sigma_k^2} (I_i - \mu_k)(I_j - \mu_k) \right) \right).$$

Here, δ_{ij} is the Kronecker delta, μ_k and σ_k^2 are the mean and variance of the intensities in the window w_k around k, and $|w_k|$ is the number of pixels in this window.

Now, since we have got the Cost function $J(\alpha)$ in terms of α only, we can define Lagrange's multiplier to optimize $J(\alpha)$

$$\alpha = \operatorname{argmin} \alpha^T L \alpha + \lambda \left(\alpha^T - b_S^T \right) D_S (\alpha - b_S),$$

where λ is some large number, D_S is a diagonal matrix whose diagonal elements are one for constrained pixels and zero for all other pixels, and b_S is the vector containing the specified alpha values for the constrained pixels and zero for all other pixels. Since the above cost is quadratic in alpha, the global minimum may be found by differentiating it and setting the derivatives to zero. This amounts to solving the following sparse linear system

$$(L + \lambda D_S) = \lambda b_S$$

5 References

1. Anat Levin, Dani Lischinski, and Yair Weiss, **A Closed-Form Solution to Natural Image Matting**, *IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 30, NO. 2, FEBRUARY 2008*

2. Chang Tang, Chunping Hou, and Zhanjie Song, **Defocus map estimation from a single image via spectrum contrast**, *Optics Letters · May 2013*

3. Shaojie Zhuo , Terence Sim**Defocus map estimation from a single image**, *Pattern Recognition Volume 44, Issue 9, September 2011*